

985714

METHOD AND APPARATUS FOR AUTOMATICALLY CHECKING E-MAIL ADDRESSES IN OUTGOING E-MAIL COMMUNICATIONS

5 FIELD OF THE INVENTION

The present invention relates to the transmission of electronic mail over computer networks, and more particularly, to a method and apparatus for checking and confirming that a valid email address is provided in an email communication prior to transmission of the communication.

10

DESCRIPTION OF THE RELATED ART

During the past decade, electronic mail ("e-mail") has become an indispensable tool for facilitating business and personal communications. Through computer networking systems such as local-area networks ("LAN"), wide-area networks ("WAN"), and the world-wide web ("WWW"), network users can send and receive notes, messages, or letters, to communicate with others who are in the same office or perhaps in remote locations across the world.

15

20

E-mail application programs are typically configured for generating messages in the form of memoranda. An e-mail application user interface guides a user to "compose" an e-mail communication by providing a platform for entering at least one outgoing e-mail address, a "subject" heading, and a "body" for the actual message. The user's return address is automatically provided in the e-mail. The user may also designate a document, file or executable program to be attached to the e-mail message. When the user completes typing the message and presses the

“send” key, the message is transmitted over the network and is routed for delivery to an e-mail server corresponding to the provided destination address.

A known e-mail communications system and a method for transmitting e-mail communications between networks over the Internet are described with reference to Fig. 1. Computers 10a-10c are connected through a local area network (LAN) 11 to e-mail communications system 12, which can send e-mail communications to any of computers 18a-18c through e-mail communications system 16 and local area network (LAN) 17. E-mail communications systems 12 and 16 include Mail Transport Agent (MTA) servers 12a, 16a, Post Office Protocol (POP or POP3) servers 12b, 16b, and Message Store 12c, 16c. The e-mail communications servers 12 and 16 are also connected to respective domain name servers (DNS) 13, 15.

When an e-mail communication is transmitted according to the Simple Mail Transport Protocol (SMTP), it is first divided into three components: the “mail from:” address; the recipient address list; and the data portion of the message. After a user of computer 10c prepares an e-mail communication and sends the e-mail across the LAN 11, it is sent to the MTA 12, which accepts e-mails for delivery. The MTA then separates the address information from the data portion of the e-mail. The MTA parses the envelope to determine whether to route the message to an external network or store the message in Message Store 12c for access by another computer connected to the LAN 11. The MTA “postmarks” the e-mail by adding routing data to the header before storing the message.

If the e-mail is to be sent to a another user on a different mail system, the MTA 12 next determines the domain for the intended recipient through the DNS 13, which queries the DNS 15 through the Internet. Upon receiving the domain information, MTA transmits the e-mail communication to the MTA 16b, which is waiting to accept e-mail. The MTA 16b then stores the received e-mail in Message Store 16c. Later, a user on computer 18a logs in to the e-mail system and connects to the POP server 16a, which determines if there is new mail to download. The POP server 16a retrieves the e-mail communication from the Message Store 16c and transmits the e-mail through the LAN 17 to the user.

As is well known, the e-mail address that is supplied by the user must be in a particular format for a successful transmission. The first part of the address is the recipient's username, followed by a "@" sign, and then a hostname or domain name, which identifies where the recipient has an Internet mail account. The rightmost portion of the domain section of the address identifies the kind of organization where the recipient has an address, which, in the United States, is typically one of: ".com", ".edu", ".gov", ".org", or ".net". During transmission, this address is translated through a domain name server by Internet Protocol (IP) into an IP address, which is a series of four numbers separated by periods (dots).

In some circumstances, e-mail software will prompt a user when an incorrect e-mail destination address is detected. For example, the software may parse the contents of an e-mail address to check that an "@" character and a domain name are present. If the e-mail address is in the wrong format, the system will prompt the

user to provide the correct e-mail address before an attempted transmission. This e-mail format checking capability can conserve network resources, while providing an immediate opportunity for a user to correct and send an e-mail communication.

It is relatively common mistake for a user to forget or misspell a username, hostname, or domain name in an e-mail address for an e-mail communication. If an invalid hostname or domain is provided, the error is detected during transmission by the domain name server. In most circumstances, an error message then generated to be transmitted back to the user. The user will then receive the error message several minutes after having attempted to send the e-mail communication. If the user exits from the e-mail software or logs off the computer network before receiving the error notification, there may be a substantial delay before the user corrects and re-sends the e-mail communication.

If a user inadvertently provides an invalid username, or an incorrect but valid hostname or domain name, the user may never be notified of the error. The e-mail will be transmitted in the usual course from the user's e-mail server, through the domain name server, and onto the Internet. The e-mail will eventually be routed to another MTA server, which will then determine that the provided username is not registered at that domain name. Although it is conceivable for the MTA server at that domain name to generate and transmit an error message in response, this is uncommon. Even if an error message is generated, it may be upwards of a half hour before this message is forwarded to the sender.

To assist the user in supplying e-mail addresses when creating e-mail communications, many e-mail software applications include an address directory. A user can generate entries by supplying names, addresses, telephone and facsimile numbers, e-mail addresses, and other pertinent information into the table. The user can later refer to the directory when composing an e-mail communication. Some e-mail software applications incorporate an automatic e-mail address generator, which, when the user provides the first few characters of an e-mail address, suggests corresponding e-mail addresses from the list that have been pre-programmed into the directory.

Although the e-mail address directory alleviates some of the above-described difficulties encountered when a user provides an incorrect e-mail address, the directory is only beneficial for those e-mail addresses that the user pre-programmed into the directory. For most users, the majority of outgoing e-mail communications are sent to persons who are not included in the user's address directory.

Accordingly, there is a need for a system for automatically generating or checking e-mail addresses to reduce the occurrence of incorrectly addressed e-mail communications.

SUMMARY OF THE INVENTION

In view of the difficulties described above with sending e-mail communications to the correct intended recipient, there is a need for a method and apparatus for automatically checking e-mail addresses provided in outgoing e-mail communications before the e-mail is transmitted over the network and along the Internet.

The present invention provides a system and method for detecting incorrect e-mail addresses in outgoing e-mail communications, and for responding by prompting a user with suggested e-mail addresses. A domain name database in the e-mail communications software creates a table of domain names by automatically storing the domain names of e-mail addresses from which incoming e-mails are received. When a user creates an outgoing e-mail communication, the system checks the domain name provided by the user with those domain names stored in the table. If the user-provided domain name does not match any of the domain names in the table, or closely resembles a domain name in the table, the user is prompted to confirm the provided e-mail address.

The e-mail communications software also tracks the number and frequency by which an e-mail address appears in incoming and outgoing e-mail communications. If an e-mail address is frequently used or received, an entry is automatically generated in a personalized e-mail address directory. The address directory is then used to check the complete e-mail addresses provided in outgoing

e-mail communications, and to suggest e-mail addresses when the system detects an incorrect e-mail address.

An object of the invention is to provide a method of automatically checking for an incorrect e-mail address in an outgoing e-mail communication. A list of domain names is stored in a memory. The list of domain names is checked for whether a domain name of the e-mail address is present in the list. A prompt is generated for a user to confirm an e-mail address if the domain name is not included in the list of domain names.

Another object of the invention is to provide a method of automatically checking for misspelled e-mail addresses in outgoing e-mail communications prior to transmission by an e-mail communications server. Domain names in senders' e-mail addresses are extracted from e-mail communications incoming to the email communications server. The extracted domain names are stored in a domain name database. Outgoing e-mail communications are received from client computers connected to the e-mail communications server through a local network. The domain name database is searched for domain names spelled similarly to the domain names in e-mail addresses provided in the outgoing e-mail communications, and an error prompt is generated upon detecting that a domain name in an e-mail address provided in an outgoing e-mail communication is misspelled.

The present invention also provides an e-mail server for automatically checking for misspelled e-mail addresses in outgoing e-mail communications prior to transmission by an e-mail communications server. The server includes an

interceptor for extracting domain names from e-mail addresses provided in incoming and outgoing e-mail communications. A domain name database stores domain names extracted from senders' e-mail addresses in incoming e-mail communications. A checker searches the domain name database for domain names spelled similarly to the domain names in e-mail addresses provided in the outgoing e-mail communications, wherein the e-mail server detects misspelled domain names in e-mail addresses in outgoing e-mail communications.

A further object of the invention is to provide a method of automatically checking for an incorrect e-mail address in an outgoing e-mail communication, comprising the steps of storing a list of e-mail addresses in a memory, checking if an e-mail address in the e-mail communication is included in the list of e-mail addresses in the memory, and generating a prompt for a user to confirm an e-mail address if the domain name is not included in the list of e-mail addresses.

Yet another object of the invention is to provide an e-mail communications system stored in a client computer for automatically checking for incorrect e-mail addresses provided in outgoing e-mail communications from the client computer prior to transmission to an e-mail server. An address extractor extracts senders' e-mail addresses from incoming e-mail communications. A memory stores e-mail addresses extracted from senders' e-mail addresses in incoming e-mail communications. A checker searches the memory for e-mail addresses that are provided in outgoing e-mail communications, The checker generates a prompt upon

detecting that an e-mail address in an outgoing e-mail communication is not present in the memory.

BRIEF DESCRIPTION OF THE DRAWINGS

5 FIG. 1 is a schematic diagram of a known e-mail communications and computer network system.

FIG. 2 is a schematic diagram of an e-mail communications server according to a first embodiment of the present invention.

10 FIG. 3 is a flow diagram for storing domain names in a domain name database in the e-mail communications server of FIG. 2.

FIG. 4 is a table of an exemplary domain name database in the e-mail communications server of FIG. 2.

15 FIG. 5 is a flow diagram for checking domain names in outgoing e-mail communications according to the first embodiment of the present invention.

20 FIG. 6. is a schematic diagram of a client-side e-mail communications system according to a second embodiment of the present invention.

FIG. 7. is a flow diagram for storing e-mail addresses in an address directory in FIG. 6 according to the second embodiment of the present invention.

FIG. 8 is a flow diagram for checking e-mail addresses in outgoing e-mail communications according to the second embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention will now be described in more detail with reference to the figures. In the e-mail communications system according to the first embodiment of the present invention, as shown in the schematic diagram provided in Fig. 2, the e-mail server includes a domain name store 20, an e-mail header interceptor/parser 21, and a domain name checker 22, in addition to the other contents of a conventional e-mail server. A Message Store 23 stores e-mail messages that are received in the e-mail communications network. MTA server 24 accepts mail or delivery and either holds it for download by a client or passes it to another MTA server for delivery. POP server 25 stores mail received via SMTP until it is read by the user.

When an e-mail communication is received in the MTA 24, e-mail communications software in the e-mail header interceptor/parser 21 scans the sender's e-mail address information, which is forwarded to the domain name checker 22. The checker 22 copies the sender's e-mail address and searches the domain name store 20 for the sender's domain name. If there is no such entry in the domain name database, a new entry is created and stored in the database 20. The checker is preferably implemented as a script file in software.

The method for populating the domain name database according to the first embodiment of the present invention is now described with reference to Fig. 3. In step 30, an e-mail communication is received and processed in the e-mail server 15. The header of the incoming e-mail message is scanned in step 31, and the domain

name address is copied in step 32. The e-mail checker 22 then searches the domain name database 20 in step 33 for the existence of the domain name copied from the incoming e-mail message. If the domain name is already stored in the domain name database, the e-mail checking operation performs any necessary database updating operations in step 34, and quits in step 35. The system may update the database by storing the date that e-mail is received, and by calculating information concerning the frequency by which e-mails having the domain are received.

If the domain name does not currently exist, a new record is created in steps 36-37. To create a new record, the database will first store the newly received domain name in step 36. Corresponding with step 34, the database may also store the date that the record is created, in step 37.

An exemplary domain name store 20 is provided in Fig. 4. The domain name store includes domain names associated with an assortment of corporate e-mail accounts and Internet service providers. If the invention is installed on a corporate e-mail system containing many e-mail accounts, and the accountholders are engaged in a variety of different business activities, there may be a wide assortment of domain names from e-mail communications that are received. As will be described, one accountholder may benefit from the domain name of an e-mail communication that was transmitted to another accountholder.

In addition to storing domain names extracted from addresses of e-mail applications that were received in the e-mail server, the domain name store may also include dates that the domain name in an e-mail address was first and most

recently received by the network, and a list that tracks the user names in the internal network who have received e-mail communications for each given domain name. The database may also calculate a “frequency factor” by which e-mail communications are received from an accountholder at the given domain name.

- 5 The “frequency factor” may be calculated according to the number of times that the domain name has been received over a certain period of time. This calculation can be weighted according to how recently the domain name has been received by the network.

10 In order to maintain the domain name database at a reasonable size, the e-mail server may be configured to periodically remove domain names from the domain name database that are used only infrequently. For example, a domain name entry may be deleted if none of the listed recipients is presently an accountholder in the network. Domain name entries also may be deleted if the “frequency factor” falls below a threshold value. Although there are several possible
15 methods for maintaining the database size, the e-mail checker will function most effectively if there are no size restrictions for the database.

20 A method for checking an outgoing e-mail communication according to the first embodiment is now described with reference to Fig. 5. A user first creates a draft e-mail communication in step 50. The e-mail communication includes at least one destination e-mail address, but may also include a subject heading, message, and/or attachment files. When the user attempts to send the e-mail communication, it is transmitted from the user’s networked computer along a LAN

and is received in the network e-mail server in step 51. The checker now extracts the domain name from the proposed recipient's e-mail address in step 52, and checks the provided domain name in step 53. If the domain name database in the e-mail server contains the provided domain name, the e-mail server processes the e-mail communication according to http protocol in step 54, and forwards the e-mail along the Internet gateway for transmission in step 55.

If the domain name database does not detect the extracted domain name in step 53, the system provides a prompt to the user in step 56. The prompt can occur in several possible formats. For example, the e-mail server may send a network message alert to the sender's personal computer, or may generate and transmit an e-mail message containing the prompt. In either case, the user can then decide in step 57 whether to confirm the e-mail address, correct the e-mail communication, or cancel delivery, in steps 58, 59, and 60, respectively. If the e-mail is confirmed in step 58, it is then processed for transmission in step 54, as described above. The system may then add the confirmed domain name to the domain name database. If the e-mail is corrected, the system will re-extract the domain name in step 52 and repeat the checking operation in step 53 before the e-mail is processed and transmitted.

In the method as described in Fig. 5 for checking an e-mail communication, the e-mail checker 22 of Fig. 2 searches the domain name database only for the domain name extracted from the user's e-mail communication, and generates a prompt when the domain name is not present in the database. As an alternative,

the checker can be configured to search for domain names that are spelled similarly to the extracted domain name. In this manner, the checker will perform a pseudo spell-checking operation to detect mistyped domain names. When a user provides an e-mail address with a domain name that is not similar to any domain names in the database, the e-mail server processes the e-mail communication, in the normal manner, without generating a prompt to the user. By only searching for words that are likely to have been mistyped or misspelled, the e-mail checker will cause less distraction for the user. Furthermore, because most e-mail addressing errors result from misspelled or mistyped domain names, this checking algorithm may be equally effective in detecting incorrect or invalid e-mail addresses.

There are numerous methods for detecting whether a user-provided domain name is similar to one or more domain names stored in the domain name database to perform a spell-checking operation. As one example, the e-mail checker can check each alpha-numeric character of the domain names in the database against those in the extracted domain name. The system can then generate a prompt when there is at least one but no more than two discrepancies between a reference domain name and the extracted domain name. The e-mail checker can also remove alpha-numeric characters from the extracted domain name, perhaps one at a time, and search the domain name database for a domain name having at least each of the remaining alpha-numeric characters. These algorithms will likely detect errors that occur when a user misspells a word by inadvertently switching two letters. The e-mail checker can also compare the extracted domain against the reference

domain names according to basic grammar rules, as is known in conventional spell-checking software.

In the first embodiment of the present invention, as specifically described with reference to Figs. 2-5, a domain name is extracted from a user-provided e-mail address and checked in the domain name database of the network e-mail server. By maintaining the domain name database at the e-mail server, as opposed to the client computers, a single domain name database becomes populated with domain names from all incoming e-mail addresses to any of the users in the network. Accordingly, the domain name database can quickly become a directory of those domain names that the users in the network often encounter. The server-side configuration of the e-mail checker and domain name database also saves processing power and storage capacity at the client-side user computers.

Although many e-mail addressing errors occur in the domain name of the e-mail address, addressing errors can also occur from misspelled or mistyped usernames at the beginning of the e-mail address. In the second embodiment of the present invention, which can be utilized separately or in conjunction with the first embodiment, the entire e-mail address is checked in a personalized e-mail address directory. The e-mail address directory, which is preferably stored on the user's client-side computer, can be incorporated into business/personal contact organizer systems that are available in many conventional e-mail communication software packages. Prior to transmitting an e-mail message composed by the user, the e-

mail system performs an e-mail address check against the database of e-mail addresses stored in the personalized address directory.

A schematic representation of the client-side e-mail communications personalized address directory and e-mail address checking system is now described with reference to Fig. 6. The client-side e-mail communications system 61 primarily consists of input/output interface 62, memory 72, e-mail messaging software 64, address directory 65, and calendar/organizer 63. The input/output interface 62 connects to the calendar/organizer 63, e-mail messaging software 64, and address directory 65 to facilitate communications with the LAN network and the user's display. As is known in the art, the e-mail messaging software component 64 includes e-mail notification software 66, templates for composing e-mail communications 67, graphical user interface 68, message and attachment viewer software 69, and file attachment software 70. The e-mail messaging software component 64 of the present invention further includes checker software 71 to detect incorrect or invalid e-mail addresses in outgoing e-mail communications, as will be explained in further detail below.

The address directory 65 includes a list of names, addresses, telephone numbers, facsimile numbers, and e-mail addresses of persons whom the user may intend to contact. The directory is logically connected to the e-mail messaging software 64 such that a user can access e-mail address information stored in the directory when composing an e-mail communication. Through a graphical user interface 73 and directory management software 74, the user may personalize the

address directory by adding new contact information, which is stored in memory 78. The address directory 65 can also be configured to communicate with other devices to assist with customization and directory maintenance, such as a user's personal digital assistant (PDA) or a business card scanner.

5 The address directory 65 of the present invention also includes address checker directory software 75 that automatically adds e-mail addresses for storage in memory 78 of address directory 65 for later reference when checking e-mail addresses in the checker 71 of e-mail messaging software 64. When the user receives an e-mail communication, the checker directory software 75 extracts the e-mail address for storage in memory 78. Alternatively, the system may be configured such that the system will first detect whether the user frequently receives e-mail communications from a particular sender before storing the e-mail address.

10 A method for populating memory 78 with e-mail addresses for e-mail checking is now described with reference to Fig. 7. In step 80, an e-mail communication is received in the user's e-mail messaging software 64. The e-mail address from the received e-mail communication is extracted by address extractor 76 in the checker directory software 75 in step 81. The system searches for the occurrence of the extracted e-mail address in memory 78 in step 82. The e-mail address may have been entered into memory 78 directly by the user via the directory management software 74, or may have been previously added by the address checker software 75. If the e-mail address is present in the memory 78, the

frequency tally 77 updates the tally associated with the e-mail address in step 83.

In this manner, frequency tally tracks how often the user receives e-mails from certain e-mail addresses. If the e-mail address is not present in the memory 78, the e-mail address is added to the memory 78 in step 84.

5 Periodically, the system checks whether the frequency tallies associated with the e-mail addresses stored in memory have fallen below a threshold level. If it is determined that the user seldom receives an e-mail communication from any given e-mail address, this e-mail address can then be deleted from memory 78, if necessary, to conserve storage space.

10 A method for checking outgoing e-mail addresses according to the second preferred embodiment is now described with reference to Fig. 8. A user drafts an e-mail communication including an outgoing e-mail address in step 90. In step 91, the e-mail address is extracted from the "to:" field in the e-mail messaging software 64 and is supplied to checker 71. The checker generates a search of the memory 78 in the address directory 65 for the supplied e-mail address in step 92. If the e-mail address is present in the address directory memory, the e-mail messaging software 64 processes the e-mail communication in step 93 and transmits the e-mail to the LAN, the e-mail server, and on to the Internet gateway in step 94.

15 If the e-mail address is not present in the address directory memory, the
20 checker 71 generates a prompt in step 95. In response to the prompt, the user can either choose in step 96 to confirm the provided e-mail address in step 97, provide a corrected e-mail address in step 98, or cancel delivery of the message in step 99. If

the user confirms that the provided e-mail address is correct, the checker then processes the e-mail communication as described with reference to step 93. If the user corrects the e-mail address, the system re-extracts the e-mail address in step 91 and repeats the checking process. This process is repeated for each e-mail address that the user provides in the "to:" and "cc:" fields of the proposed e-mail communication.

The e-mail domain name checker that operates on the e-mail server and the e-mail address checker that operates on user computers can be utilized either individually or in combination. As further alternatives, the domain name checker described with reference to Figs. 2-5 can be installed within each user computer, and/or the e-mail address checker described with reference to Figs. 6-9 can be installed on the e-mail server.

Accordingly, the e-mail domain name checker and e-mail address checker can assist the user in minimizing the occurrences of e-mail communications that would otherwise have been incorrectly sent. Thus, it is readily seen that the method and system of the present invention provides for improved and efficient e-mail communications, by detecting incorrect e-mail addresses and prompting the sender to correct the addresses prior to transmission of the e-mail communication.

The foregoing disclosure of embodiments of the present invention and specific examples illustrating the present invention have been presented for purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many variations and modifications of the

embodiments described herein will be obvious to one of ordinary skill in the art in light of the above disclosure. The scope of the invention is to be defined only by the claimed appended hereto, and by their equivalents.